

**FPGA-BASED ACCELERATOR FOR THE
IDENTIFICATION OF FINGER VEIN PATTERN VIA
K-NEAREST CENTROID NEIGHBORS**

BY

YEW TZE EE

**A Dissertation submitted for partial fulfilment of the
requirement for the degree of Master of Science
(Microelectronic Engineering)**

August 2016

ACKNOWLEDGEMENT

I have learned a lot of new skills and knowledge throughout the whole Master course. The most important criteria of doing research is keep on searching for the best method to fulfil a task. I would like to take this opportunity to acknowledge the people that made this thesis a reality.

First of all, I would like to thank my supervisor, Associate Professor Dr Bakhtiar for all the guidance and advice. Dr. Bakhtiar has taught me various theories and mentors me throughout the duration of this thesis. Dr. Bakhtiar explained sophisticated algorithm in a way student can understand easily. I would say Dr. Bakhtiar's guidance is vital and crucial to the completion of this thesis. Besides that, I also received helps and tips from my friends in making this thesis a success. Many thanks to all my friends and course mates who have lent a helping hand when I was in need of aid.

Last but not least, I would like to thank my family members especially my parents for their constant support throughout my Master dissertation period.

TABLE OF CONTENTS

	Page
ACKNOWLEDGEMENT	ii
TABLE OF CONTENTS	iii
LIST OF TABLES	v
LIST OF FIGURES	vi
LIST OF ABBREVIATIONS	viii
ABSTRAK	ix
ABSTRACT	x
CHAPTER 1 – INTRODUCTION	1
1.1 Background	1
1.2 Problem Statement	4
1.3 Research Objectives	5
1.4 Research Scope	5
1.5 Thesis Outline	6
CHAPTER 2 – LITERATURE REVIEW	8
2.1 Introduction	8
2.2 Finger Vein Recognition System in Biometrics	9
2.2.1 Criteria of a Biometric Characteristic	9
2.2.2 Finger Vein Pattern and its Advantages in Biometric System	10
2.2.3 Finger Vein Identification System	11
2.3 Classifiers	13
2.3.1 K-Nearest Neighbors (KNN) Classifier	13
2.3.2 Weighted K-Nearest Neighbors Classifier	14
2.3.3 K-Nearest Centroid Neighbors Classifier	15
2.3.4 Local Mean-Based K-Nearest Centroid Neighbors Classifier	16
2.3.5 Weighted K-Nearest Centroid Neighbors Classifier	17
2.4 FPGA-based implementation of Finger Vein Recognition System	18
2.5 Implementation of Algorithms in FPGA	20
2.6 Summary	23
CHAPTER 3 – METHODOLOGY	25
3.1 Introduction	25
3.2 Stage 1 – Development of MATLAB-based Finger Vein Identification System	26
3.2.1 Flowchart of the KNCN classifier in a software environment	27
3.3 Stage 2 – Development of FPGA-based Finger Vein Identification System	33

3.3.1 Altera Quartus II Design Software.....	35
3.3.2 Terasic DE2-115 Development Board	35
3.3.3 FPGA Design Flow	36
3.3.4 Work-around for number with fractional parts in FPGA hardware	37
3.3.5 OneNCN_FSM Module	38
3.3.6 KNCN_FSM Module	44
3.3.7 Majority_Class_FSM Module.....	51
3.3.8 RAM_Multiplexer Module	55
3.4 Stage 3 – Performance comparison between MATLAB and FPGA.....	55
3.5 Summary	56
CHAPTER 4 RESULTS AND DISCUSSION.....	58
4.1 Introduction.....	58
4.2 Accuracy of MATLAB-based Finger Vein Identification System	59
4.3 Compilation results of RTL design in Quartus	61
4.4 Simulation results of RTL design in ModelSim	62
4.4.1 OneNCN_FSM module simulation results	62
4.4.2 KNCN_FSM module simulation results	65
4.4.3 Majority_Class_FSM module simulation results	68
4.5 Accuracy of FPGA-based Finger Vein Identification System	69
4.6 Computational time of KNCN Classifier in MATLAB and FPGA	70
4.7 Summary	72
CHAPTER 5 CONCLUSION AND RECOMMENDATION	75
5.1 Conclusion	75
5.2 Recommendation for Future Research and Study.....	78
REFERENCES.....	79
APPENDICES.....	81

Appendix A: MATLAB Script of KNCN Classifier for Finger Vein Identification System

Appendix B: Top Module of KNCN Classifier in Verilog HDL for Finger Vein Identification System

Appendix C: OneNCN_FSM Module of KNCN Classifier in Verilog HDL for Finger Vein Identification System

Appendix D: KNCN_FSM Module of KNCN Classifier in Verilog HDL for Finger Vein Identification System

Appendix E: Majority_Class_FSM Module of KNCN Classifier in Verilog HDL for Finger Vein Identification System

LIST OF TABLES

	Page
Table 3.1 State Definition of FSM in OneNCN_FSM Module.....	42
Table 3.2 State Definition of FSM in KNCN_FSM Module	48
Table 3.3 State Definition of FSM in Majority_Class_FSM Module	53
Table 4.1 Accuracy of Classifiers	60
Table 4.2 Match Count of KNCN classifier in Four Sections.....	70

LIST OF FIGURES

	Page
Figure 1.1 Finger Vein Image Capturing Mechanism (Hitachi, 2006).....	3
Figure 2.1 Captured Finger Vein Pattern (Mahri, et al., 2010)	11
Figure 2.2 Flow Diagram of a General Finger Vein Identification System (Mobarakeh, et al., 2012)	12
Figure 2.3 3NCN Decision Rule - Test sample q will be assigned randomly the 3 classes (Diamond, Square and Cross) in KNCN Classifier (Grabowski, 2004).....	16
Figure 2.4 Hardware Architecture of BLPOC Method (Wei & Rosdi, 2012).....	20
Figure 2.5 Hardware Architecture of FKNN-CDM (Tan & Rosdi, 2015)	21
Figure 2.6 Hardware Architecture of PseAAC Generator (Chow, 2015).....	22
Figure 3.1 Flowchart of the Development Stages for FPGA-based Finger Vein Identification System.....	26
Figure 3.2 Flowchart for Part A of KNCN Classifier in Software Environment	29
Figure 3.3 Flowchart for Part B of KNCN Classifier in Software Environment	31
Figure 3.4 Flowchart for Part C of KNCN Classifier in Software Environment	32
Figure 3.5 Hardware Architecture of KNCN Classifier.....	34
Figure 3.6 Quartus II Design Flow	35
Figure 3.7 Top-Down FPGA Design Flow	37
Figure 3.8 Part 1 of State Machine Diagram of OneNCN_FSM Module.....	40
Figure 3.9 Part 2 of State Machine Diagram of OneNCN_FSM Module.....	41
Figure 3.10 Part 1 of State Machine Diagram of KNCN_FSM module	45
Figure 3.11 Part 2 of State Machine Diagram of KNCN_FSM Module.....	46
Figure 3.12 Part 3 of State Machine Diagram of KNCN_FSM Module.....	47
Figure 3.13 Part 1 of State Machine Diagram of Majority_Class_FSM Module	52
Figure 3.14 Part 2 of State Machine Diagram of Majority_Class_FSM.....	53
Figure 4.1 Quartus Compilation Report Summary	61
Figure 4.2 Fitter Resource Usage Summary	62
Figure 4.3 The values of Difference and Square obtained from MATLAB.....	63
Figure 4.4 The values of Difference and Square obtained from ModelSim	63

Figure 4.5 Summation of Squares and Euclidean Distance obtained from MATLAB	64
Figure 4.6 Summation of Squares and Euclidean Distance obtained from ModelSim.....	64
Figure 4.7 Euclidean Distance and Index of 1NCN in MATLAB	65
Figure 4.8 Euclidean Distance and Index of 1NCN in FPGA	65
Figure 4.9 Begin_KNCN_FSM Flag and OneNCN_Min_Index.....	65
Figure 4.10 Values of Difference and Square Obtained from MATLAB.....	66
Figure 4.11 The values of Difference and Square obtained from ModelSim.....	67
Figure 4.12 Summation of Square and Euclidean Distance obtained from MATLAB.....	67
Figure 4.13 Summation of Square and Euclidean Distance obtained from ModelSim	67
Figure 4.14 The Begin_Majority_Class_FSM Flag.....	68
Figure 4.15 Class_Reg, Class_KNCN and Training_Image_Class_80_Address_A	68
Figure 4.16 Testing_Image_Class_Match_Predicted_Class is 1	69
Figure 4.17 The Results of KNCN Classifier displayed on DE2-115.....	69
Figure 4.18 Match Count of KNCN Classifier when K equals to 9	70
Figure 4.19 Time Taken of KNCN Classifier in MATLAB	71
Figure 4.20 Number of Clock Cycles	72

LIST OF ABBREVIATIONS

BLPOC	Band-limited Phase Only Correction
FPGA	Field Programmable Gate Array
FSM	Finite State Machine
FV-USM	Finger Vein – Universiti Sains Malaysia
HDL	Hardware Description Language
IP	Intellectual Property
KNCN	K-Nearest Centroid Neighbors
KNN	K-Nearest Neighbors
KPCA	Kernel Principal Component Analysis
LE	Logic Elements
LED	Light Emitting Diode
MATLAB	Matrix Laboratory
NIR	Near-Infrared Light
PCA	Principal Component Analysis
PLL	Phase-Lock Loops
POC	Phase Only Correction
RAM	Random-access memory
ROI	Region of Interest
ROM	Read Only Memory
RTL	Register-Transfer Level
SOF	SRAM Object File
WKNCN	Weighted K-Nearest Centroid Neighbors
WKNN	Weighted K-Nearest Neighbors

PEMECUT BERASASKAN FPGA TERHADAP PENGELAS K-JIRAN TENGAH TERDEKAT UNTUK PENGECAMAN BENTUK VENA JARI

ABSTRAK

Pengecaman bentuk vena jari merupakan teknologi yang terkini dalam penyelidikan biometrik. Semua orang mempunyai corak vena jari yang berbeza. Oleh itu, bentuk vena jari merupakan satu identiti biometrik yang unik. Kebelakangan ini, para penyedik telah mengemukakan algoritma dan kaedah bagi pengecaman bentuk vena jari. Salah satu kriteria bagi pengecaman bentuk vena jari untuk menjadi teknologi biometrik yang popular ialah masa pengecaman. Kelemahan pengecaman bentuk vena jari dengan menggunakan perisian adalah peramalan program yang lambat. Untuk mengatasi kelemahan tersebut, tesis ini mencadangkan satu seni bina bagi kaedah bernama Pengelas K-Jiran Tengah Terdekat untuk mengklasifikasi bentuk vena jari di peranti *Field Programmable Gate Array* (FPGA). Pengelas K-Jiran Tengah Terdekat merupakan kaedah pengklasifikasian yang memerlukan pengiraan jiran tengah dan penyusunan k-jiran tengah terdekat. Peranti FPGA berupaya memproses maklumat secara selari berbanding dengan perisian yang memproses maklumat secara siri. Kaedah pengklasifikasian yang dilaksanakan di tesis ini direka dalam *Register Transfer Level* (RTL) dan kemudiannya diprogram ke dalam peranti FPGA Altera DE2-115 melalui Quartus II 15.0. Sebuah pengkalan data yang terdiri daripada 80 kelas imej telah digunakan untuk menilai prestasi dan ketepatan seni bina yang dikemukakan. Keputusan eksperimen memaparkan seni bina yang dikemukakan mampu memproses kaedah pengklasifikasian Pengelas K-Jiran Tengah Terdekat di peranti FPGA dalam 4.98 ms dengan ketepatan sebanyak 77 peratus yang lebih kurang sama dengan pelaksanaan berasaskan perisian. Proses peramalan kaedah Pengelas K-Jiran Tengah Terdekat di peranti FPGA lebih pantas daripada proses peramalan Pengelas K-Jiran Tengah Terdekat di perisian sebanyak sembilan kali ganda. Kesimpulannya, sistem pengecaman bentuk vena jari berasaskan Pengelas K-Jiran Tengah Terdekat telah berjaya diimplementasikan di peranti FPGA dengan masa pemprosesan yang lebih singkat berbanding dengan sistem pengecaman bentuk vena jari berasaskan perisian.

FPGA BASED ACCELERATOR FOR THE IDENTIFICATION OF FINGER VEIN PATTERN VIA K-NEAREST CENTROID NEIGHBORS

ABSTRACT

In recent years, finger vein recognition has emerged as a promising biometric technology due to the fact that each person in this world has unique finger vein pattern. Over the past few years, various finger vein recognition algorithms and techniques have been proposed by researchers and scholars. The challenge of finger vein recognition in becoming mainstream biometric technology is the processing time of recognition. As almost all finger vein recognition algorithms were implemented using software such as MATLAB which is based on general-purpose processor, the processing speed become the bottleneck of the development of finger vein recognition. In order to increase the processing, this thesis introduces an architecture for finger vein recognition based on K-Nearest Centroid Neighbors (KNCN) classifier implemented on Field Programmable Gate Array (FPGA). KNCN is a type of classification technique in image processing which involves calculation of centroid and sorting of K-nearest centroid neighbors. FPGA enables parallel computation in contrasts to serial computation in general processor. The proposed architecture is written in Verilog Hardware Description Language (HDL) and programmed into Altera Development and Educational Board, DE2-115 through Quartus II 15.0. An image database which consists of 80 finger vein classes has been used to evaluate the accuracy and processing time of the proposed architecture. Experimental results showed that the processing time of the hardware architecture for KNCN classifier implemented on FPGA for one testing image is 4.98 ms, which is 9 times faster than software implementation of KNCN classifier. The accuracy of the hardware architecture is 77% which is slightly lower than the accuracy in MATLAB. In conclusion, the finger vein identification hardware architecture based on KNCN classifier is successfully implemented on an FPGA board with faster processing time as compared with software-based implementation of KNCN classifier in MATLAB.

CHAPTER 1

INTRODUCTION

1.1 Background

Identification of human individuals, the act of finding out who someone is, has been carried out by human being since civilization. For instance, human identifies each other by face or physical appearance. There are three approaches in identification of human being and the approaches are “something you know”, “something you have” and “something you are” (Cavoukian, et al., 2012). In modern society, a popular form of identification is through unique identity card which belongs to “something you have” approach. Usually, the picture of face of an individual is in front of an identity card. The purpose of this is to facilitate the process of identification by authorities such as police or government officers. Identification by face of a person is effective because each person has unique face feature such as double eyelids, single eyelids, and so on (Dahiya & Kant, 2012). Identity card identification is a form of token-based identification (Jain, et al., 2004). Besides identity card, another popular form of identification method, knowledge-based identification is through password or pin (Bhattacharyya, et al., 2009). Knowledge-based identification belongs to “something you know” approach. For example, 6-digit pin is needed to withdraw or transfer money on ATM. Whereas password is a secret combination of either numbers or letters or both that enables a person to use a computer system such as entering a website.

As both form of identification methods mentioned, identity card and password/pin have their own weakness, another form of identification method has caught attention, which is known as biometrics (“something you are” approach). Biometrics system refers to a recognition system of individual through physiological or behavioral characteristics (Sharma, et al., 2014). Examples of biometrics system

that have been developed are fingerprint recognition, face recognition, voice recognition, signature and finger vein recognition (Wei & Rosdi, 2012).

Finger vein recognition is one of the latest method among biometrics systems which identify an individual based on the physical characteristics of their finger vein patterns (Mobarakeh, et al., 2013). Finger vein pattern refers to the network of blood vessels in human's finger which cannot be seen by human eyes (Beng & Rosdi, 2011). Finger vein pattern is totally unique for each individual and hence it is impossible to forge and falsify the finger vein identity (Damavandinejadmonfared, et al., 2012). Besides that, finger vein recognition can overcome the weakness and limitation of face recognition where glasses and make-up of individual will reduce the accuracy of face recognition system (Damavandinejadmonfared, et al., 2012). Furthermore, while face surgery can totally falsify the face recognition process, finger vein pattern of human being is unchanged over times and cannot be changed even by surgery (Mobarakeh, et al., 2012).

There are two types of finger vein recognition system, identification and verification. Finger vein identification system refers to the classification of an unknown query. For instance, if there are 80 classes of finger vein, given an unknown finger vein, which class does the finger vein belongs to? On the other hand, biometrics verification system refers to the authentication process which involves "one-to-one" match where the live biometric is being matched with the stored biometric data to validate the identity of an individual (Cavoukian, et al., 2012). Matching methods for biometrics verification system such as Band-Limited Phase-Only Correction (BLPOC), Phase-Only Correction (POC) and Local Line Binary Pattern (LLBP) are used in the finger vein research of Wei & Rosdi, (2012), Mahri et al., (2010), Asaari et al., (2014) and Rosdi et al., (2011). In this research, a hardware architecture for finger vein

identification system is proposed.

Although finger vein patterns are invisible to human eyes, finger vein patterns can be viewed through an image sensor which is sensitive to near-infrared light (NIR). The wavelengths of NIR are between 700 and 1000 nanometers (Hitachi, 2006). The near-infrared light generated from an electronics device will pass through the tissues of human fingers while blocked by pigments such as haemoglobin and melanin. Since the blood vessels are rich of haemoglobin, the near-infrared light that shines through causes the finger veins to appear as dark shadow lines in the captured finger vein image as shown in Figure 1.1. Human's finger is placed between the light source and image sensor for light transmission method of finger vein capturing mechanism. The near-infrared light passes through the finger and the NIR is then captured by the image sensor (Hitachi, 2006).

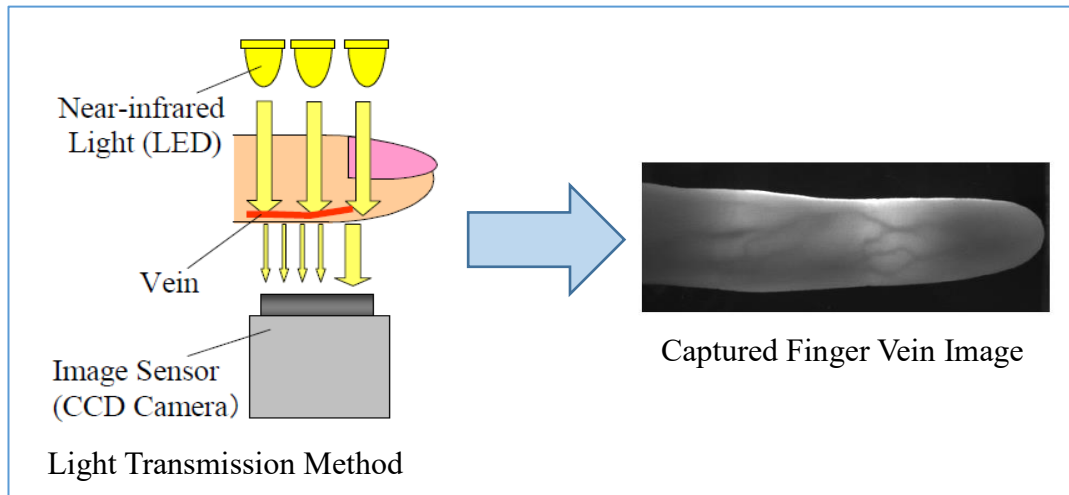


Figure 1.1 Finger Vein Image Capturing Mechanism (Hitachi, 2006)

Finger vein recognition system has several advantages over other biometrics systems which includes contactless, high accuracy and resistance to criminal tampering. Hence, an extensive research on the identification of finger vein patterns is carried out in this thesis. Classification is an important step in finger vein recognition system. Therefore, various classifiers such as K-Nearest Neighbors (KNN), Weighted

K-Nearest Neighbors (WKNN), K-Nearest Centroid Neighbors (KNCN) and Weighted K-Nearest Centroid Neighbors (WKNCN) are used for the classification of finger vein patterns in this research in MATLAB.

In previous research from other researchers, FPGA is the best candidate for high-speed computational algorithm. FPGA has been used by researchers in biometrics and bioinformatics research to reduce the processing time of computation of algorithm (Tan & Rosdi, 2015), (Wei & Rosdi, 2012). Therefore, the classifier with the highest accuracy in MATLAB is chosen to be implemented in an FPGA.

1.2 Problem Statement

For biometric technology, the database consists of all the training data set. The key factor to determine the effectiveness of an algorithm is its accuracy (percentage of correct identification count over total testing set). However, when database becomes larger as human populations keep on increasing, the speed or processing time to accomplish the identification process becomes a primary factor when choosing the technology to be implemented for a particular algorithm. This is because biometric technology such as finger vein recognition and face recognition uses complex and computational intensive image processing algorithms (Khalil-Hani & Eng, 2011). Besides accuracy and speed, the cost of the biometric system is also important because a technology which is too expensive will not be chosen by customers. Biometric technology needs to be executed on high-speed computers for reasonable processing time. Therefore, biometric technology could not be used in applications that requires low-cost embedded solutions (Khalil-Hani & Eng, 2011). Furthermore, size of a system is also crucial as bulky size system which occupies a large space will hinder the operation of the system.

In previous research carried out by other researchers, various finger vein classifiers such as Weighted K-Nearest Centroid Neighbors (WKNCN), Local Mean K-Nearest Centroid Neighbors (LMKNCN) and Fuzzy K-Nearest Centroid Neighbors (FKNCN) have been used to improve the finger vein recognition accuracy. Rosdi et al., (2015) proposed a finger vein identification using FKNCN classifier which improve the classification accuracy of finger vein recognition as compared to K-Nearest Neighbors (KNN), Fuzzy K-Nearest Neighbors (FKNN) and K-Nearest Centroid Neighbors (KNCN). Besides that, Mobarakeh et al., (2012) used WKNCN classifier in finger vein identification system. Furthermore, Mobarakeh et al., (2013) enhanced the KNCN classifier by proposing a new type of classifier called LMKNCN in finger vein classification. However, all the proposed classifiers were implemented in a software environment. This means that the accuracy of the classifiers is high but the computational speed of the proposed classifiers is slow. The implementation of KNCN classifier in FPGA can overcome the computational speed issue. This is because FPGA can perform parallel processing of computation (Tan & Rosdi, 2015).

1.3 Research Objectives

- 1) To improve the accuracy of classification for finger vein identification system using K-Nearest Centroid Neighbors classifier
- 2) To accelerate the computational speed of K-Nearest Centroid Neighbors classifier by using FPGA
- 3) To evaluate the performance of K-Nearest Centroid Neighbors classifier for finger vein identification system in MATLAB and FPGA

1.4 Research Scope

For this dissertation, a finger vein classification system based on K-Nearest

Centroid Neighbors classifier is simulated using ModelSim and implemented on an FPGA board. The primary focus of this research is on developing the hardware architecture of KNCN classifier for finger vein pattern identification system on FPGA, thus the steps in the pre-processing stage of image processing, which are image resize and normalization will be carried out in MATLAB and then the result of image normalization is used as input in FPGA. The finger vein image database used in this research is from Finger Vein – University Sains Malaysia (FV-USM) database. The input images for pre-processing in MATLAB are extracted Region of Interest (ROI) images. The architecture of KNCN classifier is designed and implemented on an FPGA board. In order to highlight the advantages of the proposed architecture, the performance of the proposed architecture is compared with implementation of KNCN classifier in MATLAB in terms of accuracy and speed.

1.5 Thesis Outline

The remainder of this thesis is organized as follows:

The second chapter is Literature Review. The purpose of literature review is to research for previous methods that are being used and carried out for the similar topics to the master thesis. After various technical papers are examined and analysed, a unique and sound approach is implemented in this dissertation.

The third chapter is about the methodology of the research, which examined the methods involved in the dissertation. The methods comprised of hardware architecture design, FPGA implementation and performance evaluation. The tools used to accomplish the research are described. Architecture design of the KNCN Classifier is also discussed. The usage and function of each component in the

architecture as well as the flow of the overall system are explained thoroughly.

The experiment results as well as analysis on the outcome are presented in Chapter Four. The results include both simulation in ModelSim and FPGA implementation in DE2-115. The performance of the proposed architecture in terms of processing time and accuracy are evaluated by making a comparison with the MATLAB-based implementation.

Chapter Five of this thesis brought the research to a conclusion and examined various enhancement and recommendation for future work. A conclusion is made based on the experimental results and analysis. Last but not least, potential improvement for future work is suggested.

CHAPTER 2

LITERATURE REVIEW

2.1 Introduction

This chapter discusses the overview of biometrics and finger vein recognition system, various classifiers for finger vein pattern identification system, implementation of finger vein recognition system in FPGA and the implementation of algorithms in FPGA. The literature review begins with an overview of biometrics criteria and finger vein recognition system. The criteria of a biometric characteristic for biometric system are discussed and the advantages of finger vein pattern in biometric system are reviewed. Biometric system is a system which recognizes individual based on their physiological and behavioral characteristics. The flow of a general finger vein identification system such as ROI extraction, image resize, image normalization and classification is explained.

As the main objective of this research is on classification of finger vein pattern, various classifiers such as KNN classifier, WKNN classifier, KNCN classifier and WKNCN classifier are studied and reviewed. Classification of finger vein pattern refers to the prediction of the class label of a testing image based on training set. The hardware implementation of BLPOC method for finger vein recognition system in FPGA is reviewed in order to examine the benefits of using FPGA to implement finger vein recognition system. BLPOC method is a matching method which is used in biometrics verification system. FPGA is a programmable device which consists of logic elements (LE), phase-locked loops (PLLs), multipliers and embedded memory that can be reconfigured to perform arithmetic calculation.

2.2 Finger Vein Recognition System in Biometrics

The criteria of a biometric characteristic for biometric system such as universality, distinctiveness, permanence and collectability are discussed. Other issues that need to be considered for a practical biometric system such as performance, acceptability and circumvention are explained thoroughly. Finger vein pattern is the network of blood vessels in human finger and it has several advantages over other biometrics such as resistant to criminal tampering, contactless and ease of feature extraction. Last but not least, the general flow of finger vein identification system is reviewed.

2.2.1 Criteria of a Biometric Characteristic

There are all sorts of biological measurements such as height, weight, DNA, fingerprints, gait, voice and finger vein. However, not all biological measurements are qualified as biometric characteristic. Jain et al., (2004) listed four requirements of human physiological and/or behavioral characteristic to be qualified as a biometric characteristic and they are:

- A. Universality: Each human being should have the physiological and/or behavioral characteristic.
- B. Distinctiveness: The physiological and/or behavioral characteristic of any two persons should be different.
- C. Permanence: The characteristic should remain unchanged over a period of time with respect to the matching criterion.
- D. Collectability: The characteristic can be measured in terms of quantity.

Besides the four criteria that were being mentioned above, there are a few matters that need to be considered for a practical biometric system, which includes:

- I. Performance: The accuracy and speed of the biometric system need to be reasonable, which means that the accuracy cannot be too low and the speed cannot be too slow. The resources that are required in order to achieve the desired recognition accuracy and speed, as well as some environmental factors and operational factors that might affect the accuracy and speed of the biometric system.
- II. Acceptability: The identifier or the characteristic of the biometric system should be accepted by public in their daily lives.
- III. Circumvention: The biometric system needs to be foolproof against fraudulent methods.

2.2.2 Finger Vein Pattern and its Advantages in Biometric System

Finger vein pattern refers to the network of blood vessels inside the finger of a human and it is invisible by human eyes. Finger vein pattern can be used as a biometric characteristic because Yanagawa et al., (2007) proved that diversity exists in human finger vein patterns of both right and left index fingers and middle fingers of human being. Besides that, finger vein patterns are unique to each person and different even among identical twins. Furthermore, finger vein patterns are time-invariant, which means that finger vein pattern remain constant throughout the adult years (Hitachi, 2006).

There are several advantages of finger vein as a biometric characteristic as compare to other biometric system:

- A. Resistant to criminal tampering: Since finger veins are inside human body, it is impossible to forge or steal the finger veins.
- B. Contactless: As human finger is placed between light source which emits near-

infrared light and an image sensor, there is no contact between human finger and light source or image sensor. Hence, the finger vein biometric system ensures cleanliness of users and also convenience of users.

- C. Ease of feature extraction: Finger vein patterns are stable and clearly captured, which means that low-resolution camera is sufficient to take vein images for small-size and simple data image processing.

2.2.3 Finger Vein Identification System

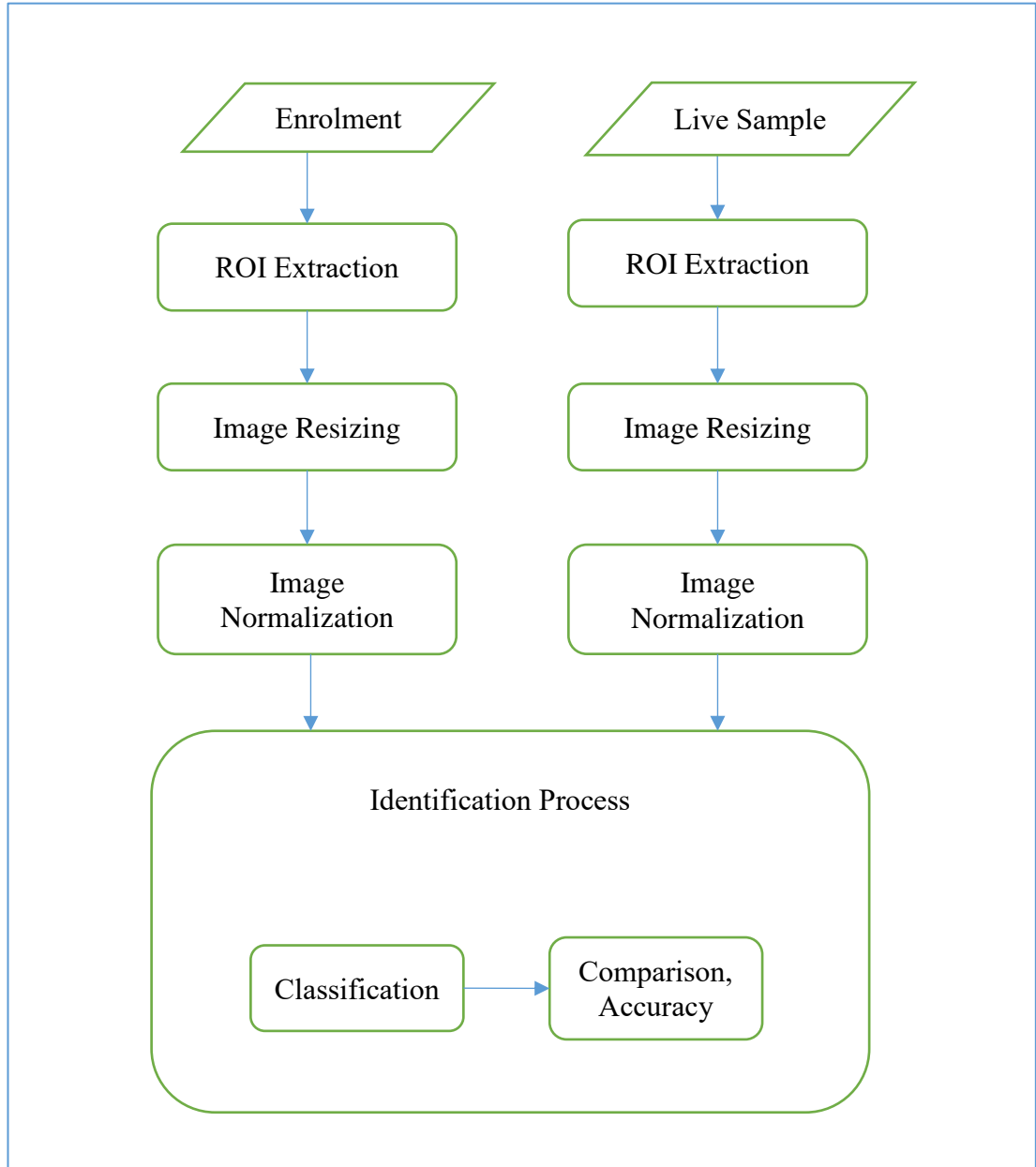
Figure 2.1 shows the capturing of finger vein of an individual and the raw finger vein pattern image is displayed on the computer. The captured finger vein images are stored in a database and this process is called the enrolment process as shown in Figure 2.2. The live sample is the testing image of finger vein pattern. The raw finger vein pattern image captured contains some black unwanted background which will reduce the finger vein recognition accuracy (Mahri, et al., 2010). Hence, extraction of region of interest (ROI) needs to be done on the original finger vein image (Mobarakeh, et al., 2012).



Figure 2.1 Captured Finger Vein Pattern (Mahri, et al., 2010)

The extracted finger vein images are then resized and normalized as shown in Figure 2.2. The finger vein images are resized to a smaller size in order to overcome time complexity as well as eradicate pixel noise. Image normalization on finger vein images is needed to increase the accuracy of the classification. After image resizing

and image normalization, classification is done on the finger vein image of the live sample where a class label will be assigned to the query based on the algorithm of the classifier. The classification of finger vein image is correct if the class label predicted by the classifier matches the actual class of the live sample.



*Figure 2.2 Flow Diagram of a General Finger Vein Identification System
(Mobarakeh, et al., 2012)*

Weighted K-Nearest Centroid Neighbor (WKNCN) classifier is proposed by Mobarakeh, et al., (2012) using Kernel Principle Component Analysis (KPCA) for

feature extraction for finger vein recognition system. Besides that, Local Mean based K-Nearest Centroid Neighbor (LMKNCN) classifier for finger vein recognition is proposed by Mobarakeh, et al., (2013) to improve the accuracy of traditionally used method like KNN classifier in finger vein identification system. KNN classifier is a classification method and has been used in finger vein identification system. The drawback of KNN classifier is the existence of outliers will greatly reduce the classification accuracy in small training sample size (Mobarakeh, et al., 2013). The experiments carried out by Damavandinejadmonfared, et al., (2012) show that LMKNCN classifier outperform KNN classifier in terms of classification accuracy.

2.3 Classifiers

Classifiers can be categorized into parametric method and non-parametric method. Parametric method employs probability models whereas non-parametric method determines the decision boundaries directly. K-Nearest Neighbors (NN) classifier is a type of non-parametric method in classification system (Kumar & Sahoo, 2012). K-Nearest Neighbors (KNN) classifier and its variants, Weighted K-Nearest Neighbors (WKNN) classifier, K-Nearest Centroid Neighbors (KNCN) classifier, Local Mean-based K-Nearest Centroid Neighbors (LMKNCN) classifier and Weighted K-Nearest Centroid Neighbors (WKNCN) classifier will be explained in the following subsections.

2.3.1 K-Nearest Neighbors (KNN) Classifier

KNN classifier is one of the classification method in machine learning which classify objects or points based on nearest or closest training samples in a given feature space. The classification of an object in KNN classifier is through majority vote of its

neighbors (Tamouk & Allahakbari, 2012). The object will be assigned the class with highest occurrence of the classes among its k-nearest neighbors where k is a positive integer. The value of K is determined by the user which depends on the situation where the classification method is used.

After the value of K in KNN classifier is determined by the user, Euclidean distance between query point and all training points is calculated and the results are stored (Tamouk & Allahakbari, 2012). The Euclidean distance between a query point called x , where $x = (x_1, x_2, \dots, x_n)$ and its neighbors x_i^{NN} , where $x_i^{NN} = (x_i^{NN_1}, x_i^{NN_2}, \dots, x_i^{NN_n})$ in Euclidean n-space can be found by Equation (2.1) (Gou, et al., 2012). After the Euclidean distance between query point and each training points are computed, the training point with the shortest Euclidean distance is assigned as 1NN (the nearest neighbor). The training point with the second shortest Euclidean distance is assigned as 2NN (the second nearest neighbor). Similarly, the training point with K^{th} shortest Euclidean distance will be assigned as KNN.

$$\begin{aligned} d(x, x_i^{NN}) &= \sqrt{(x_1 - x_i^{NN_1})^2 + (x_2 - x_i^{NN_2})^2 + \dots + (x_n - x_i^{NN_n})^2} \\ &= \sqrt{\sum_{q=1}^n (x_q - x_i^{NN_q})^2} \end{aligned} \quad (2.1)$$

2.3.2 Weighted K-Nearest Neighbors Classifier

For KNN classifier, all nearest neighbors are weighted equally which means that they all have a weightage of 1. In Weighted K-Nearest Neighbors (WKNN) classifier, only the nearest neighbor with K equals to 1 has a weightage of 1. Dudani, (1976) introduced a distance-weighted K-Nearest Neighbors (WKNN) classifier with the concept of weighting neighbors which are closer to the query more heavily. This means that the closer the neighbor to the query, the higher the weightage of that

neighbor. This method is less sensitive to the choices of the size of neighborhood as compared to KNN classifier (Gou, et al., 2012).

If the number of training samples, N in a training set is very large as compared to the number of nearest neighbors, K considered (i.e. $N \gg K$), the outcome of the two classifiers, KNN classifier and WKNN classifier will be comparable to each other. However, if the training set is small or moderate size, the use of WKNN classifier will yield smaller probabilities of error (Dudani, 1976). The weightage of K-Nearest Neighbors is calculated according to Equation (2.2). The weightage of the K-Nearest Neighbors equals to 1 when K equals to 1.

$$w_i = \frac{d(x, x_K^{NN}) - d(x, x_i^{NN})}{d(x, x_K^{NN}) - d(x, x_1^{NN})}, \text{ if } d(x, x_K^{NN}) \neq d(x, x_1^{NN}) \quad (2.2)$$

2.3.3 K-Nearest Centroid Neighbors Classifier

While K-Nearest Neighbors classifier only concern about the Euclidean distance between the training points and query point, K-Nearest Centroid Neighbors requires that the neighbors to be symmetrically distributed around the query point (Chaudhuri, 1996). In Figure 2.3, 1NCN (the nearest centroid neighbor), 2NCN (the 2nd nearest centroid neighbor) and 3NCN are well distributed around the query point while 1NN (the nearest neighbor), 2NN (the second nearest neighbor) and 3NN (the third nearest neighbor) are all concentrated on the left of the query point. This means that 1NN, 2NN and 3NN in Figure 2.3 are three closest points or nearest neighbors to the query point. However, they are not symmetrical around the query point. KNCN classifier is an enhancement to KNN classifier in the sense that the neighbors are symmetrical to the query point.

small training size settings. For a practical classifier, the classifier should be robust to outliers (Mitani & Hamamoto, 2006). The main objective of using Local Mean-Based K-Nearest Centroid Neighbors Classifier (LMKNCN) is to overcome the negative effect of existing outliers in the training set (Gou, et al., 2012). Besides that, LMKNCN classifier is proposed to handle the problem that existed in KNCN classifier which is the ties of vote in majority voting for making a decision (Gou, et al., 2012).

The main difference between LMKNCN classifier and KNCN classifier is that LMKNCN classifier finds the Euclidean distance between the local mean vector of k-nearest centroid neighbors from each class and the query after computing the nearest centroid neighbors whereas KNCN classifier classifies the query through majority voting. The local mean vector, u is calculated according to Equation (2.3) where r is the number of nearest centroid neighbors for a particular class and i represents each class. Next, LMKNCN classifier will assign the class label with the lowest Euclidean distance between the query and the local mean vector to the query.

$$u_{ir}^{NCN} = \frac{1}{r} \sum_{j=1}^r x_{ij}^{NCN} \quad (2.3)$$

2.3.5 Weighted K-Nearest Centroid Neighbors Classifier

Similar to KNN classifier, KNCN classifier assumes that k-nearest centroid neighbors of a query have equal weight when classify the query. This implicit assumption is not suitable in pattern classification (Du, et al., 2012). This is because the tie of votes in majority voting of class in KNCN classifier can give rise to undesirable classification results. In other words, different weightage for each k-nearest centroid neighbor is necessary for a good classification algorithm.

As the nearest centroid neighbors in KNCN are chosen based on the Euclidean distance between centroid and query, there are situations where the Euclidean distance

of the i^{th} centroid neighbor is smaller than the $(i-1)^{\text{th}}$ centroid neighbor. The distance-weighted function proposed by Dudani, (1976) will result in negative weightage of the nearest centroid neighbors. Hence, it is not appropriate to use the distance-weighted function in KNCN. The solution to this problem is to weight the reliable centroid neighbors more and reduce the weightage of unreliable centroid neighbors (Du, et al., 2012).

There are two formulas to calculate the weightage of k-Nearest Centroid Neighbors, Heat kernel function and uniform kernel function (Du, et al., 2012). WKNCN based on Heat kernel function is known as Heat Weighted K-Nearest Nearest Centroid Neighbors (HWKNCN) while WKNCN based on uniform kernel function is known as Uniform Weighted K-Nearest Centroid Neighbors (UWKNCN). UWKNCN method performs slightly better than HWKNCN method in experiments carried out by Du et al., (2012). The weightage for each centroid neighbor are calculated based on Equation (2.4) where i indicates the rank of K nearest centroid classifiers and ranges from 1 to K. For example, the weightage of 1NCN (w_1^{NCN}) is 1, the weightage is 2NCN (w_2^{NCN}) is $\frac{1}{2}$ which equals to 0.5 and the weightage of 3NCN (w_3^{NCN}) is $\frac{1}{3}$ which equals 0.33.

$$w_i^{NCN} = \frac{1}{i} \quad (2.4)$$

2.4 FPGA-based implementation of Finger Vein Recognition System

Prior to this research, there is no research on FPGA-based KNCN classifier on finger vein identification system being published. Therefore, review could not be made on this particular topic. Fortunately, there are a few references on FPGA-based implementation of finger vein recognition system and will be discussed in this section. Wei & Rosdi, (2012) implemented the BLPOC method on hardware for verification of

finger vein pattern of an individual. The hardware implementation of the finger vein recognition system in FPGA by BLPOC method is proposed in order to overcome the problem on the processing speed of software-based finger vein recognition system. The processing time of the proposed hardware architecture for BLPOC method on a single matching is 2.7ms, whereas the processing time of software-based finger vein recognition system for BLPOC method is 26ms (Wei & Rosdi, 2012). Thus, the acceleration of the computational speed of BLPOC method is 9 times faster in FPGA.

BLPOC method is an enhanced method of Phase-Only Correction (POC) method. POC method is a matching method which uses the phase components in Two Dimension Discrete Fourier Transform (2D-DFT) of input images to check on their similarity. BLPOC method is a form of POC method which limits the band of phase components in order to remove most of the meaningless phase components from matching. The accuracy of the finger vein recognition system based on BLPOC method is increased after the meaningless phase components are removed from matching. The hardware architecture of BLPOC method for finger vein recognition system is shown in Figure 2.4 where Discrete Fourier Transform (DFT) is the main block (Wei & Rosdi, 2012). The DFT coefficients of the input image is calculated in the DFT block. The hardware architecture is written in Verilog HDL and simulated in Altera Quartus II using USM database.

The implementation of BLPOC method for finger vein recognition system on FPGA by Wei & Rosdi, (2012) is for verification of an individual. Finger vein verification system is a one-to-one finger vein recognition system where the live sample of finger vein pattern of an individual is being matched with the stored finger vein pattern data of the same individual. On the other hand, finger vein identification system is a one-to-many finger vein recognition system which recognizes an individual

by searching the stored finger vein pattern data of all individuals in the database for classification (Jain, et al., 2004). The classifier will assign a class label to the live sample based on its algorithm. Finger vein identification system in FPGA is developed in this research in order to enhance the finger vein recognition system.

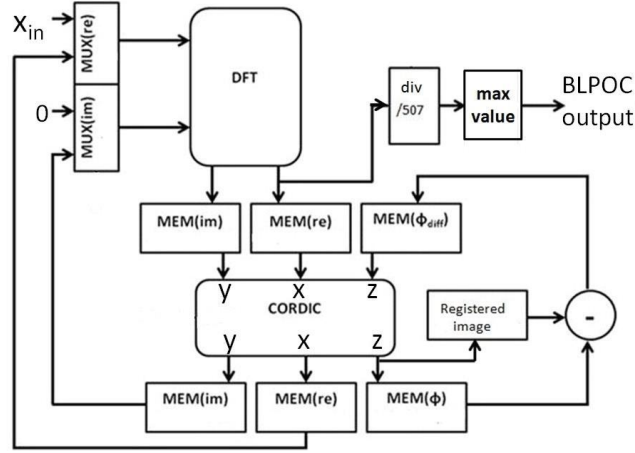


Figure 2.4 Hardware Architecture of BLPOC Method (Wei & Rosdi, 2012)

2.5 Implementation of Algorithms in FPGA

Besides the implementation of finger vein recognition system in FPGA, FPGA is also used by researchers to accelerate the processing speed for computational heavy algorithms. One of the field that requires fast processing speed for computation of algorithm is bioinformatics. FPGA has been utilized in bioinformatics extensively in recent years. For instance, FPGA was used as an accelerator for the prediction of protein secondary class using fuzzy K-Nearest Neighbors (FKNN) in Tan & Rosdi, (2015) research as the software-based implementation of the prediction mechanism requires lengthy computational time.

As the Lempel-Ziv (LZ) decomposition algorithm in Fuzzy K-Nearest Neighbors-Complexity based Distance Measure (FKNN-CDM) suffers from high computational load due to computational complexity increases exponentially with the length of the protein sequence, a hardware architecture is developed by Tan & Rosdi,

(2015) to speed up the computation process of the FKNN-CDM algorithm due to its parallel data processing capability. The FKNN-CDM developed for the prediction of protein secondary class incorporates fuzzy set theory into Nearest Neighbors-Complexity based Distance Measure (NN-CDM) in order to overcome uncertainty issue and increase the prediction accuracy. The hardware architecture designed by Tan & Rosdi, (2015) for the prediction of protein secondary class via FKNN-CDM is shown in Figure 2.5 and implemented on Altera DE2-115 board with an acceleration of 15 times faster in terms of computational speed as compared to its software counterpart. Hence, the advantage of FPGA is to speed up the computational process of an algorithm. The fast computational speed of FPGA is beneficial to the finger vein identification system in this research as the performance of biometric system is important for a practical biometric system as discussed in Section 2.2.1.

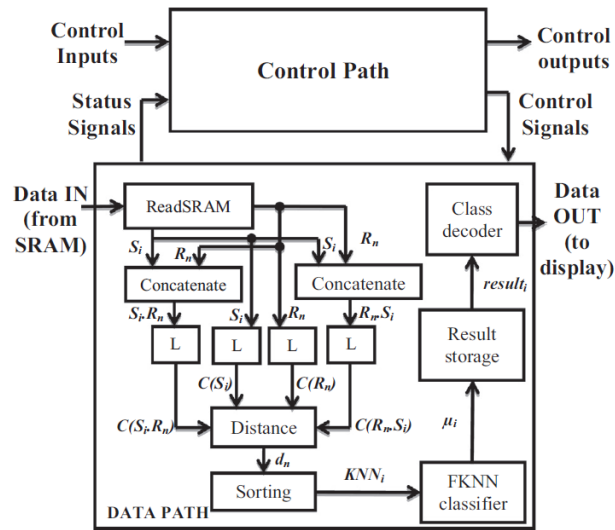


Figure 2.5 Hardware Architecture of FKNN-CDM (Tan & Rosdi, 2015)

Besides that, FPGA was used to implement Pseudo-Amino Acid Composition (PseAAC) modelling technique which incorporates sequence-order information of a protein into a discrete model selectively. An FPGA-based PseAAC generator can overcome the limitation of PseAAC webserver and PseAAC-Builder as it doesn't

require internet connection and can process the algorithm in shorter time through parallel processing. The FPGA-based PseAAC generator hardware architecture developed by Chow, (2015) improves the computational speed by 31.5 times over a Perl-based PseAAC generator. The key of the improvement lies in two computational-intensive modules, Sum-of-Small-T module and T-u-minus-20 module are run in parallel in order to capitalize on parallel processing capability of the FPGA (Chow, 2015). The capability of FPGA to accelerate the computational process of an algorithm can greatly enhance the finger vein identification system of this research. The processing time of classification of finger vein pattern of an individual can be shorten significantly by using FPGA. Hence, FPGA is used in this research for the implementation of finger vein identification system to speed up the classification of finger vein image. The hardware architecture of the PseAAC generator developed by Chow, (2015) is shown in Figure 2.6.

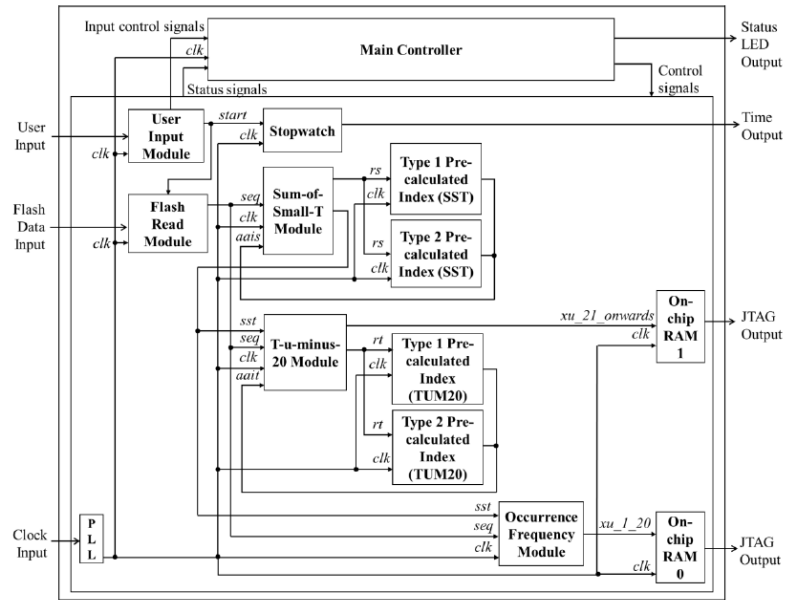


Figure 2.6 Hardware Architecture of PseAAC Generator (Chow, 2015)

2.6 Summary

In this chapter, a brief overview of finger vein recognition system is given and various classifiers for classification as well as the usage of FPGA in different fields are discussed. As not all biological measurements can be qualified to be a biometric, criteria of a biometric characteristic such as universality and distinctiveness are examined thoroughly. Finger vein pattern of an individual emerged to be a promising biometric feature as it has several merits such as contactless, ease of feature extraction and resistant to criminal tampering. A general finger vein identification system is discussed where ROI extraction, image resizing and image normalization are performed on enrolment and live sample before the identification process. Classification of the live sample is then carried out and accuracy of the algorithm is determined.

Next, five classifiers for classification are discussed in details. KNN classifier is one of the simplest classification method in machine learning field where the concept of distance between query and training points is used. The class label for the query is assigned based on the majority voting of the class of the nearest neighbors. WKNN classifier is introduced by Dudani, (1976) in order to rectify the problem that arises in KNN classifier which weights all neighbors equally. KNCN classifier is proposed by Chaudhuri, (1996) in such a way that the neighbors need to be both close to the query and symmetrically distributed around the query. LMKNCN classifier and WKNCN classifier enhance the KNCN classifier by local mean vector and different weightage of nearest centroid neighbors respectively.

Lastly, the implementation of finger vein recognition system using FPGA and the usage of FPGA in bioinformatics are discussed. FPGA is a programmable device which has parallel processing capability in contrast to software in general processor

which executes the code in a program sequentially. For instance, two modules can be executed in parallel using FPGA as demonstrated in Chow, (2015) research.